



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

LOG3000 – INGÉNIERIE LOGICIELLE

Examen intra

Présenté à:

Mohammad Hamdaqa

Écrit par :

Cassy Charles (1947025)

Remis le 25 Février 2022

Code d'honneur

En répondant à cet examen à domicile ("take-home") de LOG3000, je m'engage à :

- avoir complété cet examen par moi-même, sans l'aide d'autres personnes ;
- ne remettre que mes propres travaux, qui ne contiennent aucun matériel plagié ;
- ne divulguer pas les questions posées dans cet examen et les réponses aux questions à quiconque ne suit pas LOG3000 cette session ;
- ne pas participer à des activités malhonnêtes visant à améliorer mes résultats ou à nuire aux résultats d'autres personnes.

Signature : Cassy Charles

Date et lieu : Montréal, le 25 février 2022

I. Question sur l'utilité des processus (1 point)

Présentez deux utilités des processus de développement logiciel (0.5 point).

Deux utilités des processus de développement logiciel sont :

La communication interne qui permet à l'équipe de comprendre comment le logiciel est développé, à quoi sert leur travail.

L'évaluation des pratiques en place : Diagnostic qui permet d'évaluer les problèmes dans les manières actuelle de développement du logiciel.

Justifiez chaque utilité en présentant pour chacune un contexte où cette utilité serait importante (0.5 point).

Un contexte où cette utilité serait importante :

a) Communication interne

À travers une réunion SCRUM, un échange sur le développement du logiciel, selon le travail de chaque individu dans l'équipe, est tenu. Ce qui permet aux employés de comprendre le développement du logiciel, quelles informations et ressources sont disponibles et le rôle de chacun.

b) Évaluation des pratiques en place : Diagnostic :

Un gestionnaire de projet se rend compte qu'un processus de type cascade qui permet de déployer des nouvelles versions du logiciel chaque année introduisait beaucoup de bogues. Il détecte un problème dans la manière de développement du logiciel.

II. Question sur le choix d'un cycle de vie (4 points)

Pour les quatre contextes ci-dessous, expliquez (avec une justification brève) quel cycle de vie de base (incrémental, transformationnel, spiral ou cascade) serait le plus approprié :

a) Le développement du logiciel pour une nouvelle voiture complètement autonome (driverless car). (1 point)

Le cycle de vie approprié pour cette méthode est Spirale car c'est un projet complexe à haut risque. Pour la construction de cette voiture, de nouvelles technologies seront utilisées et que c'est en rapport avec les normes. Vu qu'elle est autonome, il faut s'assurer que le contrôle ait un faible niveau d'erreur, ce qui correspond à l'analyse de risque du modèle spirale.

- b) Le développement d'une application de livraison de colis suivant l'application d'algorithmes du plus court chemin pour assurer la livraison dans les meilleurs délais. (1 point)

Le cycle de vie approprié pour cette méthode est incrémental car c'est un projet avec un faible niveau de risque et un faible niveau d'incertitudes. Il est vrai que l'algorithme du plus court chemin est connu. Il se peut que les routes changent, et qu'on veuille utiliser un algorithme du plus court chemin différent, et il faudra ajuster l'application en conséquent. Avec ce modèle, il y a plus de flexibilité et cette gestion de risque est plus facile à gérer.

- c) Le développement d'un système d'assistance pour la réservation d'un workspace qui fonctionne dans les téléphones intelligents, les tablettes, les ordinateurs et les téléviseurs intelligents. (1 point)

Le cycle de vie approprié pour cette méthode est transformationnel car les exigences sont floues. Il est juste spécifié qu'il faut développer un système d'assistance pour une réservation. Il faut juste faire des cycles rapides ou on peut faire des corrections et recevoir du feedback avec des utilisateurs pour l'ajuster. L'inconvénient est qu'il peut finir à la poubelle et coûter plus cher.

- d) Le développement d'une application Web qui synthétise des informations sur le nombre de personnes vaccinées par région et par pays. (1 point)

Le cycle de vie approprié pour cette méthode est cascade car les exigences sont connues et le risque à prendre est très minime. Il n'y a pas beaucoup d'aspects novateurs car l'affichage de l'application Web change seulement en fonction des données des personnes. L'application n'est pas sujette à un changement. De ce fait, le cycle de vie cascade est le plus approprié.

III. Question sur la gestion du risque (4 points)

- a) Pour chacun des risques, identifiez s'il s'agit d'un risque direct ou d'un risque indirect. (1 point)

R1 : Risque indirect

R2 : Risque direct

R3 : Risque direct

- b) Proposez une approche de gestion des risques pour chaque risque identifié. Dans votre réponse, décrivez brièvement ce qui peut être fait spécifiquement. (Par exemple, si vous choisissez la stratégie « mitiger » pour un risque, vous devez décrire ce qui peut être fait spécifiquement pour mitiger le risque.) (3 points)

R1 : une approche de gestion de risque réactive serait de contourner. L'équipe n'est pas sûre du fonctionnement de la librairie de reconnaissance d'images choisie pour le client. Ainsi, si le prototype avec cette librairie ne marche pas, il faut changer et réessayer avec une autre librairie.

R2 : une approche de gestion de risque proactive serait d'éviter. L'équipe peut réorganiser leur code et éviter d'utiliser l'algorithme de neurones et se pencher sur l'utilisation d'un autre algorithme plus simple à leur niveau qui ne relève pas nécessairement de l'intelligence artificielle.

R3 : une approche de gestion de risque serait « sans-action ». Vu que le niveau du risque est faible, l'équipe peut vivre avec le code mal structuré, mal documenté et difficile à comprendre. Il se peut qu'il ne soit pas réutilisé. Ce qui pousse à penser que l'approche sans-action est la meilleure dans cette situation.

IV. Question sur la discipline de gestion de projet (3 points)

- a) **En tant que gestionnaire de projet, à quel endroit est-ce que vous vous situeriez dans le triangle de fer ? Quel(s) pôle(s) serai(en)t votre priorité ? Justifiez votre choix (1 point).**

En tant que gestionnaire de projet, je me situerais plus au niveau de la qualité car ce projet vise à créer un appareil numérique de défibrillation cardiaque. Je veux produire un produit de qualité meilleure car il doit pouvoir redémarrer le cœur ou de régler un problème d'arythmie cardiaque. Par conséquent, soit le développement de l'application va prendre du temps ou sera coûteuse. On ne connaît pas l'échéancier du projet ni le budget.

- b) **Décrivez un cycle de vie qui serait compatible avec l'endroit que vous avez choisi dans le triangle (1 point).**

Un cycle de vie compatible avec l'endroit choisi est le modèle cascade. Ce modèle décrit une faible incertitude au niveau des besoins du client avec une réalisation technique. Peu d'aspects novateurs y sont associés avec une faible participation du client. Il décrit aussi une faible urgence de voir le besoin fini, ce qui correspond à la position choisie dans le triangle qui s'éloigne du respect de l'échéancier.

- c) **Justifier votre choix d'approche (cycle de vie) de développement (1 point).**

Les exigences des appareils numériques de défibrillation cardiaques sont déjà connues, car on sait qu'on veut analyser l'activité cardiaque d'une personne en arrêt cardio-respiratoire à l'aide d'électrodes posées sur la peau et de produire automatiquement des défibrillations afin de redémarrer le cœur ou de régler un problème d'arythmie cardiaque. La réutilisation répétitive de cet appareil serait judicieuse. Les inconvénients de cette approche sont le dépassement d'échéanciers, de budgets. Ce qui n'importe pas trop dans le cadre de notre projet vu qu'on se situe au niveau de la qualité et qu'on accepte que le projet prenne du temps et qu'il soit coûteux. Ce qui nous porte à penser que le modèle cascade est approprié.

V. Question sur la qualité des exigences (3 points)

Partie A : Dans le contexte d'un service logiciel qui vise à prévenir les habitants du Québec par cellulaire, télévision et radio lors des calamités, indiquez pour chacune des exigences ci-dessous si l'exigence est fonctionnelle ou non-fonctionnelle. Expliquez brièvement votre réponse.

- a) Le système doit permettre aux responsables de la sécurité québécoise de créer des alertes. (0,75 point)

Fonctionnelle : car cette exigence décrit une fonctionnalité du logiciel. Cela peut être construit en une tâche définie qui est de créer des alertes. La manière dont on veut les alertes n'est nullement spécifiée.

- b) Le système doit envoyer des messages dans un format accessible aux personnes aveugles ou incapables de lire. (0,75 point)

Non-fonctionnelle : cette exigence décrit une qualité de logiciel qui est d'avoir un format accessible aux aveugles. Ceci est une description de la manière dont on veut que les aveugles reçoivent les messages qui est le « format accessible ».

Partie B : Les exigences suivantes pour ce logiciel ont des problèmes. D'abord, identifiez les problèmes sur chaque exigence. Ensuite, réécrivez chaque exigence pour corriger les problèmes.

- a) Le système doit être très fiable. (0.75 point)
- L'exigence doit être testable : Le mot fiable ne décrit pas clairement la fiabilité du système.
 - ❖ Le système doit être fonctionnel 99% du temps.
- b) Le système doit distribuer les alertes sur plusieurs plateformes et les autorités doivent être en mesure de contrôler la distribution des alertes. (0.75 point)

- Exigence composée : deux exigences dans une phrase
- Utilisation du pluriel : il y a ambiguïté pour représenter les alertes et les autorités
 - ❖ Le système doit distribuer une alerte **sur une plateforme**.
 - ❖ Le système doit permettre à une autorité de contrôler la distribution d'une alerte.

VI. Question sur l'analyse qualitative de processus (4 points)

Comme tout processus, le choix de vos collègues a des forces et des faiblesses.

a) Décrivez trois (3) problèmes auxquels vos collègues sont exposés en suivant ce processus (3 points).

- i. Il n'y a aucune phase d'analyse et de conception. Le respect du cycle de développement du modèle incrémental n'est pas respecté. L'implémentation du code commence directement après avoir priorisé les exigences, ce qui peut causer des problèmes d'architecture et un code mal structuré.
- ii. L'activité remue-méninges permet d'arriver à une liste d'exigences importantes. Or, les besoins initiaux ne sont pas clairs et il manque une activité pour mieux comprendre les exigences. Il est nécessaire de faire une étude des cas d'utilisation des acteurs et des parties prenantes pour éviter des problèmes plus tard dans le développement.
- iii. Un autre problème est quand l'équipe trouve des défauts après les tests d'intégration. Elle devrait retourner dans la phase d'implémentation. Au lieu de cela, des nouvelles exigences ou des corrections d'exigences sont faites. Il manque l'artéfact de validation du code testé sans problèmes avant l'ajustement des exigences.

b) Décrivez, textuellement et/ou à travers un modèle, une correction au processus qui permettrait de mieux gérer un des problèmes identifiés précédemment. Vous n'avez à proposer qu'une correction. Vous devez cependant justifier votre correction. (1 point).

Pour le problème (ii), une correction au processus serait de définir une architecture et une conception pour les différentes composantes du projet et spécifier les interactions entre les classes. Ce qui évitera une ambiguïté lors de l'implémentation.

VII. Question sur l'analyse quantitative de processus (5 points)

a) En utilisant les données de la figure, identifiez et décrivez le cycle de vie utilisé pour le développement de ce nouveau logiciel (1 point).

Le cycle de vie utilisé pour le développement de ce logiciel est incrémental. Ce cycle comprend généralement des itérations typiques de 2 à 6 semaines et dans notre cas, ils font des itérations de 4 semaines. De plus, dans ce modèle, chaque itération se concentre sur une fraction des exigences. Les itérations reprennent les exigences, l'analyse, l'implémentation et les tests. Ce qu'on remarque dans notre cas avec la reprise des exigences, de l'analyse, du code et des tests toutes les quatre semaines.

b) En utilisant le cycle de vie et les données de la figure, décrivez des problèmes potentiels concernant la maintenance (2 points).

- i. Des problèmes d'architecture peuvent survenir dans ce problème, vu que tous les requis ne sont pas connus dès le départ. De plus, on observe qu'à la semaine 22, les phases d'analyse et de conception ont été arrêtés. Donc, ils ont implémenté du code sans la phase de design, qui peut causer des problèmes de maintenance du code.

- ii. Un autre problème serait de constater que peu de tests sont effectués vers la fin du projet, jusqu'à ce qu'aucun ne soit plus effectué, à partir de la semaine 25. Ce qui peut causer un projet de qualité médiocre.

c) En tant que gestionnaire de l'équipe de maintenance, et tenant compte des problèmes décrits précédemment, que feriez-vous afin de mitiger les défis de maintenance liés à ce nouveau logiciel (2 points) ?

En tant que gestionnaire de l'équipe de maintenance, pour mitiger les défis de maintenance, il serait bon d'analyser le projet, de revoir l'architecture du code, pour évaluer l'impact et le niveau de risque pour voir si ça vaut la peine de refactoriser le code. De plus, des tests unitaires et des tests d'intégration sur le nouveau changement seraient effectués. Les bogues trouvés seraient corrigés et retestés pour s'assurer de la bonne qualité du logiciel.

VIII. Question sur la théorie de la mesure (5 points)

a) Vous analysez une équipe distribuée de conception et de développement de logiciels. Tous les membres de l'équipe travaillent à distance. Le but de l'analyse est de s'assurer que les membres de l'équipe collaborent de manière productive. En utilisant l'approche GQM, identifiez deux questions pour ce but ; puis identifiez une métrique quantitative pour chaque question (3 points).

Deux questions pour ce but :

- i. Est-ce que les tâches assignées aux membres de l'équipe respectent l'échéancier ?
 - ❖ Ratio du temps requis pour effectuer une tâche vs le temps demandé pour la tâche
- ii. Est-ce que les membres de l'équipe planifient bien leurs tâches sur l'outil Jira ?
 - ❖ Temps requis pour effectuer une tâche

b) Vous menez une étude utilisateur pour comprendre l'utilisabilité d'un système de réservation d'activités hivernales. Identifiez l'échelle de mesure pour les métriques suivantes (2 points) :

- i. L'heure de la journée à laquelle la réservation est effectuée (par exemple, 10h20, 11h10, 12h50 etc.).
Intervalle
- ii. Le nombre d'erreurs que l'utilisateur fait lorsqu'il complète une tâche de réservation en ligne. **Ratio**
- iii. La satisfaction globale de l'utilisateur du système (en cinq niveaux : très insatisfait, insatisfait, neutre, satisfait, très satisfait). **Ordinale**
- iv. Le type d'activité que l'utilisateur a recherché au cours de l'étude utilisateur (par exemple, ski, glissades, traineaux, etc.). **Nominale**

IX. Réflexion (1 point)

Réfléchissez aux sujets du cours que nous avons couvert jusqu'à présent. Énumérez les sujets qui, selon vous, sont les plus utiles (0,5 point) et les sujets dont vous souhaitez en savoir plus, mais qui manquent dans le cours (0,5 point).

Les sujets qui me sont le plus utiles sont les étapes du développement logiciel telles que la discipline des requis, de l'analyse et de la conception, de l'implémentation des tests.

Il serait bien d'en apprendre plus sur le développement agile et les différentes étapes de ce développement comme l'approche UPEDU, car c'est ce qu'on applique dans la vie réelle. Il serait important de parler plus du document d'exigences, vu qu'il remplit une partie importante dans le développement d'un logiciel. Aussi, il serait bon de parler plus du modèle de cycle de vie linéaire.